

In the Claims

1. (Currently amended) A framework for developing software applications, comprising:

a document manager operable to manage activities relating to one or more XML-based data documents associated with an application, each data document modeling data associated with the application and representing relationships that may exist between the data document and one or more other data documents;

a rules manager operable to manage rules that specify logic for the handling of notifications associated with XML-based action documents received at the application, the action documents defining notifications supported by one or more applications developed using the framework, the notifications defined by the action documents requesting actions involving data documents; and

an operations manager operable to manage execution of transactions involving data documents based on one or more operations defined for the application during its development;

the framework providing a generic XML-based transaction engine, the application being one of a plurality of distributed applications each being an instance of the generic transaction engine.

2. (Original) The framework of Claim 1, wherein a data document may be related to another data document using an XML-based tag specified during definition of the data document.

3. (Original) The framework of Claim 1, wherein at least one data document is remote from the application, the application operable to access the data document using an address specified during deployment of the application, the application operable to determine the address for the data document substantially on the fly during operation of the application.

4. (Original) The framework of Claim 1, wherein the action documents are defined to expose application program interfaces (APIs) used to notify data documents.

5. (Original) The framework of Claim 1, wherein each rule comprises a condition and one or more action statements, the application operable to:

- receive a notification;
- apply each associated rule to the notification;
- execute the action statements if the condition is satisfied; and
- return an error if the condition is not satisfied.

6. (Original) The framework of Claim 5, wherein an action statement is selected from the group consisting of:

- an action statement causing a state transition;
- an action statement setting an attribute of an associated document;
- an action statement causing a document to be acted upon;
- an action statement causing a notification to be sent to an application;
- an action statement causing a message to be sent; and
- an action statement causing another rule to be invoked.

7. (Original) The framework of Claim 1, wherein each operation is packaged in an XML-based request.

8. (Original) The framework of Claim 7, wherein the request comprises at least one parent data document and one or more child data documents.

9. (Original) The framework of Claim 8, wherein the parent data document comprises a purchase order and the child data documents comprise line items of the purchase order.

10. (Original) The framework of Claim 1, wherein an operation is selected from the group consisting of creating a document, retrieving a document, modifying a document, and deleting a document.

11. (Original) The framework of Claim 1, wherein one or more of the operations are made available through the framework during definition of associated data documents to provide a rich XML interface for interacting with these data documents.

12. (Original) The framework of Claim 11, wherein the framework provides access to software development capabilities for programming for particular needs if the operations provided through the framework are insufficient to meet those needs.

13. (Original) The framework of Claim 1, further comprising a services manager that manages transfer of XML-based information between the application and one or more other distributed applications.

14. (Original) The framework of Claim 13, wherein the services manager helps provide a peer-to-peer architecture allowing the application to transparently access services and data associated with other distributed applications.

15. (Currently amended) An XML-based software application, comprising:

- a document manager operable to manage activities relating to one or more XML-based data documents associated with the application, each data document modeling data associated with the application and representing relationships that may exist between the data document and one or more other data documents;
- a rules manager operable to manage rules that specify logic for the handling of notifications associated with XML-based action documents received at the application, the action documents defining notifications supported by the application, the notifications defined by the actions documents requesting actions involving data documents; and
- an operations manager operable to manage execution of transactions involving data documents based on one or more operations defined for the application during its development;
- the application developed using an XML-based framework providing a generic XML-based transaction engine, the application being one of a plurality of distributed applications each being an instance of the generic transaction engine.

16. (Original) The application of Claim 15, wherein a data document may be related to another data document using an XML-based tag specified during definition of the data document.

17. (Original) The application of Claim 15, wherein at least one data document is remote from the application, the application operable to access the data document using an address specified during deployment of the application, the application operable to determine the address for the data document substantially on the fly during operation of the application.

18. (Original) The application of Claim 15, wherein the action documents are defined to expose application program interfaces (APIs) used to notify data documents.

19. (Original) The application of Claim 15, wherein each rule comprises a condition and one or more action statements, the application operable to:

- receive a notification;
- apply each associated rule to the notification;
- execute the action statements if the condition is satisfied; and
- return an error if the condition is not satisfied.

20. (Original) The application of Claim 19, wherein an action statement is selected from the group consisting of:

- an action statement causing a state transition;
- an action statement setting an attribute of an associated document;
- an action statement causing a document to be acted upon;
- an action statement causing a notification to be sent to an application;
- an action statement causing a message to be sent; and
- an action statement causing another rule to be invoked.

21. (Original) The application of Claim 15, wherein each operation is packaged in an XML-based request.

22. (Original) The application of Claim 21, wherein the request comprises at least one parent data document and one or more child data documents.

23. (Original) The application of Claim 22, wherein the parent data document comprises a purchase order and the child data documents comprise line items of the purchase order.

24. (Original) The application of Claim 15, wherein an operation is selected from the group consisting of creating a document, retrieving a document, modifying a document, and deleting a document.

25. (Original) The application of Claim 15, wherein one or more of the operations are made available through the framework during definition of associated data documents to provide a rich XML interface for interacting with these data documents.

26. (Original) The application of Claim 15, further comprising a services manager that manages transfer of XML-based information between the application and one or more other distributed applications.

27. (Original) The application of Claim 26, wherein the services manager helps provide a peer-to-peer architecture allowing the application to transparently access services and data associated with other distributed applications.

28. (Currently amended) An electronic marketplace, comprising:
a distributed transaction layer; and
a plurality of distributed XML-based software applications developed using an XML-based framework which provides a generic XML-based transaction engine, each application being an instance of the generic transaction engine, the applications operable to interact with one another using the distributed transaction layer to conduct electronic commerce within the marketplace, each application comprising:

a document manager operable to manage activities relating to one or more XML-based data documents associated with the application, each of the data documents modeling data associated with the application and representing relationships that may exist between the data document and one or more other data documents;

a rules manager operable to manage rules specifying logic for handling notifications associated with XML-based action documents received at the application, the action documents defining notifications supported by the application, the notifications defined by the action documents requesting actions involving data documents; and

an operations manager operable to manage the execution of transactions involving data documents based on one or more operations defined for the application during its development.

29. (Original) The marketplace of Claim 28, wherein the distributed transaction layer provides communications between the applications in XML over HTTP format.

30. (Original) A method of developing a software application using an XML-based application development framework, comprising:

using the framework, providing a document manager to manage activities relating to one or more XML-based data documents associated with the application, each data document modeling data associated with the application and representing relationships that may exist between the data document and one or more other data documents;

using the framework, providing a rules manager to manage rules that specify logic for the handling of notifications associated with XML-based action documents received at the application, the action documents defining notifications supported by the application developed using the framework, the notifications defined by the action documents requesting actions involving data documents; and

using the framework, providing an operations manager to manage execution of transactions involving data documents based on one or more operations defined for the application during its development;

the framework providing a generic XML-based transaction engine, the application being one of a plurality of distributed applications each being an instance of the generic transaction engine.

31. (Original) The method of Claim 30, wherein a data document may be related to another data document using an XML-based tag specified during definition of the data document.

32. (Original) The method of Claim 30, wherein at least one data document is remote from the application, the application accessing the data document using an address specified during deployment of the application, the application determining the address for the data document substantially on the fly during operation of the application.

33. (Original) The method of Claim 30, wherein the action documents are defined to expose application program interfaces (APIs) used to notify data documents.

34. (Original) The method of Claim 30, wherein each rule comprises a condition and one or more action statements, the application:

- receiving a notification;
- applying each associated rule to the notification;
- executing the action statements if the condition is satisfied; and
- returning an error if the condition is not satisfied.

35. (Original) The method of Claim 34, wherein an action statement is selected from the group consisting of:

- an action statement causing a state transition;
- an action statement setting an attribute of an associated document;
- an action statement causing a document to be acted upon;
- an action statement causing a notification to be sent to an application;
- an action statement causing a message to be sent; and
- an action statement causing another rule to be invoked.

36. (Original) The method of Claim 30, wherein each operation is packaged in an XML-based request.

37. (Original) The method of Claim 36, wherein the request comprises at least one parent data document and one or more child data documents.

38. (Original) The method of Claim 37, wherein the parent data document comprises a purchase order and the child data documents comprise line items of the purchase order.

39. (Original) The method of Claim 30, wherein an operation is selected from the group consisting of creating a document, retrieving a document, modifying a document, and deleting a document.

40. (Original) The method of Claim 30, wherein one or more of the operations are made available through the framework during definition of associated data documents to provide a rich XML interface for interacting with these data documents.

41. (Original) The method of Claim 40, further comprising providing access to software development capabilities for programming for particular needs if the operations provided through the framework are insufficient to meet those needs.

42. (Original) The method of Claim 30, further comprising using the framework to provide a services manager to manage transfer of XML-based information between the application and one or more other distributed applications.

43. (Original) The method of Claim 42, wherein the services manager helps provide a peer-to-peer architecture allowing the application to transparently access services and data associated with other distributed applications.

44. (Currently amended) A method of conducting electronic commerce using a plurality of XML-based software applications, comprising:

using a document manager of a first application associated with an electronic marketplace, managing activities relating to one or more XML-based data documents associated with the first application, each data document modeling data associated with the first application and also representing relationships that may exist between the data document and one or more other data documents;

using a rules manager of the first application, managing rules that specify logic for the handling of notifications associated with XML-based action documents received at the first application, the action documents defining notifications supported by the first application, the notifications defined by the action documents requesting actions involving data documents; and

using an operations manager of the first application, managing the execution of transactions involving data documents based on one or more operations defined for the first application during its development;

the first application developed using a framework providing a generic XML-based transaction engine, the first application being one of a plurality of distributed applications each being an instance of the generic transaction engine.

45. (Original) The method of Claim 44, wherein a data document may be related to another data document using an XML-based tag specified during definition of the data document.

46. (Original) The method of Claim 44, wherein at least one data document is remote from the application, the application accessing the data document using an address specified during deployment of the application, the application determining the address for the data document substantially on the fly during operation of the application.

47. (Original) The method of Claim 44, wherein the action documents are defined to expose application program interfaces (APIs) used to notify data documents.

48. (Original) The method of Claim 44, wherein each rule comprises a condition and one or more action statements, the application:

- receiving a notification;
- applying each associated rule to the notification;
- executing the action statements if the condition is satisfied; and
- returning an error if the condition is not satisfied.

49. (Original) The method of Claim 48, wherein an action statement is selected from the group consisting of:

- an action statement causing a state transition;
- an action statement setting an attribute of an associated document;
- an action statement causing a document to be acted upon;
- an action statement causing a notification to be sent to an application;
- an action statement causing a message to be sent; and
- an action statement causing another rule to be invoked.

50. (Original) The method of Claim 44, wherein each operation is packaged in an XML-based request.

51. (Original) The method of Claim 50, wherein the request comprises at least one parent data document and one or more child data documents.

52. (Original) The method of Claim 51, wherein the parent data document comprises a purchase order and the child data documents comprise line items of the purchase order.

53. (Original) The method of Claim 44, wherein an operation is selected from the group consisting of creating a document, retrieving a document, modifying a document, and deleting a document.

54. (Original) The method of Claim 44, wherein one or more of the operations are made available through the framework during definition of associated data documents to provide a rich XML interface for interacting with these data documents.

55. (Original) The method of Claim 44, further comprising using a services manager to manage transfer of XML-based information between the application and one or more other distributed applications.

56. (Original) The method of Claim 55, wherein the services manager helps provide a peer-to-peer architecture allowing the application to transparently access services and data associated with other distributed applications.

57. (Currently amended) An XML-based software application, comprising:
means for managing activities relating to XML-based data documents associated with the application, each data document modeling data associated with the application and representing relationships that may exist between the data document and one or more other data documents;

means for managing rules specifying logic for handling notifications associated with XML-based action documents that are received at the application, the action documents defining notifications supported by the application, the notifications defined by the action documents requesting actions involving data documents; and

means for managing execution of transactions involving data documents, based on one or more operations defined for the application during its development;

the application developed using an XML-based framework providing a generic XML-based transaction engine, the application being one of a plurality of distributed applications each being an instance of the generic transaction engine.